

BAB II

TINJAUAN PUSTAKA

II.1. Sistem

Secara sederhana suatu sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari suatu unsur, komponen, atau variabel yang terorganisir, saling tergantung satu sama lain dan terpadu. Teori sistem secara umum yang pertama kali diuraikan oleh Kennet Boulding, terutama menekankan pentingnya perhatian terhadap setiap bagian yang membentuk sebuah sistem. Kecenderungan manusia yang mendapat tugas memimpin suatu organisasi adalah terlalu memusatkan perhatian pada salah satu komponen saja dari sistem organisasi.

Menurut Scott (1996) sistem terdiri dari unsur- unsur seperti masukan (input), pengolahan (processing), keluaran (output) (Hanif Al Fatta 2007:4). Menurut McLeod (2004) sistem adalah sekelompok elemen-elemen yang terintegrasi dengan tujuan yang sama untuk mencapai tujuan (Yakub, 2012:1).

Sedangkan menurut Jogiyanto (1999) terdapat dua kelompok pendekatan sistem didalam mendefinisikan sistem yaitu pendekatan pada prosedur dimana sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, terkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk tujuan tertentu, dan pendekatan pada komponen- komponen atau elemen-elemen, pendekatan pada komponen dianggap lebih mudah dalam mempelajari sistem untuk tujuan dan perancangan system (Dani Eko Hendrianto, 2014:58).

Dari kesimpulan diatas, Sistem adalah entitas atau satuan yang terdiri dari dua atau lebih komponen atau subsistem (sistem yang lebih kecil) yang saling terhubung dan terkait untuk mencapai suatu tujuan.

II.1.1. Konsep Dasar Sistem

Sistem (system) dapat didefinisikan dengan pendekatan prosedur dan dengan pendekatan komponen. Dengan pendekatan prosedur, sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu. Contoh sistem yang didefinisikan dengan pendekatan prosedur ini adalah sistem akuntansi. Sistem ini didefinisikan sebagai kumpulan dari prosedur-prosedur penerimaan kas, pengeluaran kas, penjualan, pembelian dan buku besar. Dengan pendekatan komponen, sistem dapat didefinisikan sebagai kumpulan dari komponen yang saling berhubungan satu dengan yang lainnya membentuk satu kesatuan untuk mencapai tujuan tertentu (Alfredo Ornai da Graça Cruz, 2012:4).

Contoh sistem ini didefinisikan dengan pendekatan ini misalnya adalah sistem komputer yang didefinisikan sebagai kumpulan dari perangkat keras dan perangkat lunak.

II.1.2. Karakteristik Sistem

Suatu sistem mempunyai karakteristik. Karakteristik sistem adalah sebagai berikut :

1. Komponen Sistem (components system)
2. Batas Sistem (boundary system)

3. Lingkungan Luar Sistem (environments system)
4. Penghubung Sistem (interface system)
5. Masukan Sistem (input system)
6. Keluaran Sistem (output system)
7. Pengolah Sistem (process system)
8. Sasaran Sistem (objectives system) (Alfredo Ornai da Graça Cruz, 2012:5).

II.2. Informasi

Pada Informasi adalah data yang diolah menjadi bentuk yang berguna dan lebih berarti bagi yang menerimanya. (1) Akurat (*Accurate*) berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Akurat juga berarti informasi harus jelas mencerminkan maksudnya. (2) Tepat waktu (*Timelines*) berarti informasi yang datang pada si penerima tidak boleh terlambat. (3) Relevan (*Relevance*) berarti informasi bermanfaat bagi pemakainya. diklasifikasikan diolah atau diinterpretasi untuk digunakan dalam proses mengambil keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya mengolah dari bentuk tak berguna menjadi berguna bagi penerimanya (Deppi Linda, 2012:62)

Informasi adalah data yang telah diklasifikasikan atau diinterpretasi untuk digunakan dalam mengambil keputusan (Deppi Linda, 2012:52).

II.3. Sistem Informasi

Sistem informasi adalah berupa suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan data transaksi harian yang mendukung operasi yang bersifat managerial dengan kegiatan strategi suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan (Dani Eko Hendrianto, 2014:58).

Berikut ini adalah definisi sistem informasi menurut beberapa ahli:

Tabel II.1. Definisi Sistem Informasi

Sumber	Definisi
Mefadden, dkk (1999)	Mendefinisikan Informasi sebagai data yang telah diproses sedemikian rupa sehingga meningkatkan pengetahuan seseorang yang menggunakan data tersebut. Shanon dan Weaver, dua orang insinyur listrik, melakukan pendekatan secara matematis untuk mendefinisikan informasi
Kroenke, (1992)	Menurut mereka, Informasi adalah “jumlah ketidakpastian yang dikurangi ketika sebuah pesan diterima”. artinya, dengan adanya informasi, tingkat kepastian tidak meningkat.
Davis, (1999)	Sistem Informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau saat mendatang.
Burch dan Grudnitski, (1989)	Sistem Informasi menggambarkan pengelolaan data menjadi informasi dan pemakaian informasi untuk mengambil keputusan, hingga akhirnya dari tindakan hasil pengambilan keputusan tersebut dihasilkan data kembali.

(Sumber : *Alfredo Ornai da Graça Cruz, 2012, Hal : 5*)

II.4. Perpustakaan

Banyak batasan atau pengertian perpustakaan yang disampaikan oleh para pakar dibidang perpustakaan. Ada beberapa pengertian perpustakaan dibawah ini :

- a. Perpustakaan diartikan sebuah ruangan atau gedung yang digunakan untuk menyimpan buku dan terbitan lainnya yang biasanya disimpan menurut tata susunan tertentu yang digunakan pembaca bukan untuk dijual (Sulistya Basuki, Pengantar Ilmu Perpustakaan, Universitas Terbuka, 1993).
- b. Dalam pengertian yang terbaru seperti yang tercantum dalam keputusan Presiden RI Nomor II disebutkan bahwa perpustakaan merupakan salah satu sarana pelestarian bahan pustaka sebagai teknologi dan kebudayaan dalam rangka mencerdaskan kehidupan bangsa dan menunjang pelaksanaan pembangunan nasional (Marto Atmojo Karmidi, Manajemen Perpustakaan Khusus, Universitas Terbuka, Jakarta, 1997).
- c. Menurut RUU Perpustakaan pada Bab I pasal 1 menyatakan Perpustakaan adalah institusi yang mengumpulkan pengetahuan tercetak dan terekam, mengelolanya dengan cara khusus guna memenuhi kebutuhan intelektualitas para penggunanya melalui beragam cara interaksi pengetahuan.

Secara umum dapat disimpulkan bahwa pengertian perpustakaan adalah “suatu institusi unit kerja yang menyimpan koleksi bahan pustaka secara sistematis dan mengelolanya dengan cara khusus sebagai sumber informasi dan dapat digunakan oleh pemakainya”. Sistem Informasi Perpustakaan menurut Gordon B.davis (2003) : “Sistem Informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan data harian, penunjang

kegiatan dalam penyimpanan data, dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.”

“Sistem Informasi Perpustakaan (SIPERPUS) merupakan perangkat lunak yang didesain khusus untuk mempermudah pendataan koleksi perpustakaan, katalog, data anggota/peminjam, transaksi dan sirkulasi koleksi perpustakaan” (Dani Eko Hendrianto, 2014:58)

II.4.1. Jenis-jenis Perpustakaan

Berdasarkan keputusan Mendikbud tentang pokok-pokok kebijakan pembinaan dan pengembangan perpustakaan di Indonesia ada beberapa jenis perpustakaan antara lain:

1. Perpustakaan nasional
2. Perpustakaan wilayah
3. Perpustakaan umum
4. Perpustakaan sekolah
5. Perpustakaan perguruan tinggi (Alfredo Ormai da Graca Cruz,2012:6).

II.4.2. Fungsi perpustakaan

Perpustakaan sebagai salah satu lembaga yang berperan aktif dalam peningkatan sumber informasi dan peningkatan sumberdaya alam, sangatlah penting artinya dalam usaha mencerdaskan kepentingan bangsa. Perpustakaan dilambangkan sebagai tempat bertanya dalam sumber informasi tentang ilmu pengetahuan yang sifatnya khusus maupun umum. Jadi secara umum tujuan

perpustakaan pada waktu sekarang ini ialah agar setiap orang yang datang ke perpustakaan mencari informasi dan kebutuhan-kebutuhan ilmu pengetahuan tidak akan pulang dengan tangan hampa, tetapi pasti akan mendapat segala apa yang dibutuhkan (Alfredo Ornai da Graça Cruz, 2012:6).

II.5. Database

Database merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan diperangkat keras komputer dan digunakan diperangkat lunak untuk memanipulasinya (Jogiyanto HM : 1999:711). *Database* merupakan kumpulan data yang saling berelasi. Data sendiri merupakan salah satu komponen yang sangat penting dalam sistem informasi, karena merupakan basis sistem dalam menyediakan informasi bagi para pemakai (Deppi Linda, 2012:63).

Menurut (Asrianda, 2008), Database adalah sekumpulan tabel-tabel yang saling berelasi, relasi tersebut bisa ditunjukkan dengan kunci dari tiap tabel yang ada. Satu database menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi.

Database mempunyai kegunaan dalam mengatasi penyusunan dan penyimpanan data, maka seringkali masalah yang dihadapi adalah:

1. Redundansi dan Inkonsistensi data
2. Kesulitan dalam pengaksesan data
3. Isolasi data untuk standarisasi
4. Multi user
5. Keamanan data

6. Integritas data

7. Kebebasan data (Gellysa Urva dan Helmi Fauzi Siregar, 2015:93)

II.6. SQL Server 2008

SQL Server 2008 adalah sebuah RDBMS (*Relational Database Management System*) yang sangat *powerful* dan telah terbukti kekuatannya dalam mengolah data. Dalam versi terbarunya ini, SQL Server 2008 memiliki banyak fitur yang bisa diandalkan untuk meningkatkan performa *database*. SQL Server 2008 memiliki suatu GUI (*Graphic User Interface*) yang kita gunakan untuk melakukan aktivitas sehari-hari berkaitan dengan *database*, seperti menulis T-SQL, melakukan *backup* dan *restore database*, melakukan *security database* terhadap aplikasi, dan sebagainya. Pada GUI tersebut kita bisa melakukan settingan terhadap SQL Server untuk berkerja lebih optimal. *Settingan* juga bisa dilakukan menggunakan *script* untuk memudahkan developer mengubah *Setting Options* pada SQL Server 2008. (Ruslan, 2013:39).

II.7. Microsoft Visual Studio 2010

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap yang dapat digunakan untuk melakukan pengembangan aplikasi, baik itu aplikasi bisnis, aplikasi *personal*, ataupun komponen aplikasi lainnya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Kompiler yang dimasukkan ke dalam paket *Visual Studio* antara lain *Visual C++*, *Visual C#*,

Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe (Ahmad Rais Ruli Skom, 2017:10).

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di atas *Windows*) ataupun *managed code* (dalam bentuk *Microsoft Intermediate Language* di atas *.NET Framework*). Selain itu, *Visual Studio* juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*). *Visual Studio* kini telah menginjak versi *Visual Studio 12.0* atau dikenal dengan sebutan *Microsoft Visual Studio 2013* yang diluncurkan pada 17 Oktober 2013 yang ditujukan untuk *platform Microsoft .NET Framework 4.5.1*. Versi sebelumnya *Visual Studio 2012* ditujukan untuk platform 4.5, *Visual Studio 2010* ditujukan untuk *.NET Framework 4.0*, *Visual Studio 2008* ditujukan untuk *platform .NET Framework 3.5*, *Visual Studio 2005* ditujukan untuk *platform .NET Framework 2.0* dan 3.0. *Visual Studio 2003* ditujukan untuk *.NET Framework 1.1*, dan *Visual Studio 2002* ditujukan untuk *.NET Framework 1.0*. Versi-versi tersebut di atas kini dikenal dengan sebutan *Visual Studio .NET*, karena memang membutuhkan *Microsoft .NET Framework*. Sementara itu, sebelum muncul *Visual Studio .NET*, terdapat *Microsoft Visual Studio 6.0 (VS1998)* (Ahmad Rais Ruli Skom, 2017:10).

Beberapa kemampuan atau dari *Visual Basic* diantaranya adalah:

1. Untuk membuat program aplikasi berbasis *Windows*.
2. Untuk membuat objek-objek pembantu seperti misalnya *Control ActiveX*, *file help*, aplikasi *internet* dan sebagainya.

3. Menguji program (*Debugging*) dan menghasilkan program akhir berakhiran *exe* yang bersifat *executable* atau dapat langsung dijalankan (Ahmad Rais Ruli S,kom, 2017:11).

VB 6 dan VB.NET mempunyai beberapa perbedaan mendasar, yaitu:

1. Integrated Development Environment (IDE). VB.NET masih menggunakan Visual Studio sebagai IDE, namun sudah berbeda dengan Visual Studio yang digunakan pada VB6, sekarang Visual Studio hanya mendukung tiga bahasa utama yaitu VB, C#, dan C++.
2. Perubahan sintaks dan model objek dari class. VB.NET sudah didesign ulang menjadi lebih modern maka banyak sintaks yang dikurangi dan ditambahkan, misal perintah GoSub sudah dihilangkan tapi banyak keyword baru terutama untuk Object Oriented Programming.
3. Perubahan kompilasi kode dan menjalankan program. Perbedaan yang ketiga terletak pada proses untuk kompilasi dan menjalankan program. VB.NET application akan dikompilasi menjadi kode assembly tidak seperti kebanyakan aplikasi Win32. .NET juga memiliki komponen Garbage Collection yang akan secara otomatis membersihkan objek yang dibuat dari memori ketika sudah tidak diperlukan. Jadi tidak perlu menghapus objek secara manual (Ahmad Rais Ruli S,kom, 2017:11).

VB.NET adalah salah satu bahasa pemrograman tingkat tinggi yang mendekati bahasa manusia. Kemunculan bahasa VB.NET ini sebagai jawaban untuk menyederhanakan bahasa pemrograman pada platform .NET yang diluncurkan tahun 2002 dan untuk menjembatani programmer Visual Basic.

Bahasa VB.NET secara teknis mengadopsi sintak bahasa Visual Basic. Konsistensi API membuat bahasa VB.NET menjadi pilihan dalam membuat kode program diatas platform Windows.

Fitur baru bahasa VB.NET dibandingkan Visual Basic bahwa bahasa VB.NET mendukung object-oriented dan juga dynamics programming. Ini menambah daftar kemudahan untuk belajar bahasa VB.NET (Agus Kurniawan,2013:10).

II.8. UML (*Unified Modelling Language*)

Unified Modeling language merupakan suatu bahasa pemodelan untuk melakukan spesifikasi, visualisasi, konstruksi dan dokumentasi objek dalam pengembangan sebuah perangkat lunak atau sistem yang berfungsi untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa real world terdiri dari object- object dasar yang mempunyai hubungan atau relasi antar object- object tersebut.

UML hanya berfungsi untuk melakukan pemodelan. Penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek.

Bahasa Pemodelan *UML* sering digunakan untuk pembuatan perangkat lunak dalam bahasa pemrograman berorientasi objek namun demikian tetap dapat digunakan pada bahasa pemrograman prosedural (Siti Nurajizah,2015:215).

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language (UML)*. *UML*

adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak.


UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. *UML* saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem. (Gellysa Urva dan Helmi Fauzi Siregar, 2015, Hal : 93).

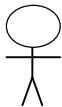


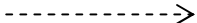
Dalam pembuatan Tugas Akhir ini penulis menggunakan diagram *Use Case* yang terdapat di dalam UML. Adapun maksud dari *Use Case* Diagram diterangkan dibawah ini.

II.8.1. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. *Use Case Diagram*

Gambar	Keterangan
	<i>Use case</i> menggambarkan fungsionalitas yang

	<p>disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p>
	<p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.</p>
	<p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi</p>




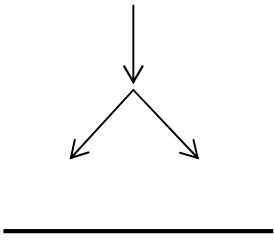
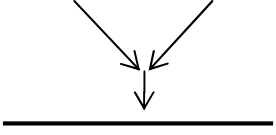
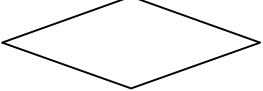

	program.
←-----	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

Sumber : (Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 94)

II.8.2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.3 dibawah ini:

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End point</i> , akhir aktivitas.
	<i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	<i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> .
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

Sumber : (Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 94)

II.8.3. *Class Diagram*

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut.

Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini :

Tabel II.4. *Multiplicity Class Diagram*

Multiplicity	Penjelasan
--------------	------------

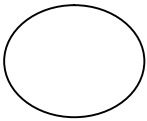
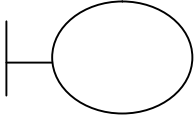
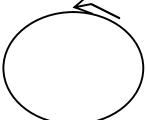
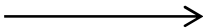
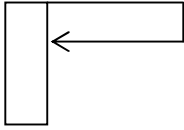
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4



Sumber : (Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 95)

II.8.4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.5 dibawah ini :

Tabel II.5. Simbol Sequence Diagram

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar <i>class</i> .
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.

	<p><i>Activation, activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p>
	<p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p>

Sumber : (Gellysa Urva dan Helmi Fauzi Siregar, 2015 : 95)